# **Desafio: Construindo um Pipeline de Dados com Python**

# **🚀 Desafio**

<aside> 📎 **Arquivos do Desafio:**

* **Python** para processamento e transformações de dados.
* **Apache Airflow** para orquestrar o pipeline.
* **Docker** para executar o Airflow em um contêiner, isolando suas dependências e ambiente:
* **Base de dados:** [raw\_data.csv](https://prod-files-secure.s3.us-west-2.amazonaws.com/6a055055-52ec-4ebb-a697-63027c951344/109745a7-935b-4e00-9635-834d94ee4913/raw_data.csv)

</aside>

## **Contexto:**

Na era da transformação digital, dados precisos são essenciais para empresas como a "DncInsight Solutions", especializada em análise e processamento de dados. A empresa enfrenta desafios com a qualidade e organização dos dados recebidos, muitas vezes inconsistentes e incompletos. Para resolver isso, "DncInsight Solutions" iniciou um desafio interno para desenvolver um sistema robusto e automatizado para o processamento de dados.

Como engenheiro de dados na "DncInsight Solutions", você é responsável por desenvolver um pipeline de dados usando Apache Airflow. Este pipeline transformará dados brutos em insights valiosos através de um processo que inclui a limpeza e agregação de dados. Diferentemente da abordagem tradicional com AWS S3, você simulará um ambiente de produção usando pastas locais para armazenamento de dados.

## **Como começar?**

Sua tarefa é projetar e implementar um pipeline de dados que ingere dados brutos, os processa e os armazena em três camadas distintas (bronze, prata e ouro) . O pipeline deve ser orquestrado usando o Apache Airflow. Lembre-se, o Windows não suporta o Airflow, sugerimos a utilização de Docker para executar o Airflow.

# **🎯 Etapas de Desenvolvimento**

## **Etapa 01) Configuração Inicial**

Configure o Apache Airflow usando Docker para criar um ambiente uniforme e controlado. Esta configuração inicial garante consistência nas dependências e facilita o gerenciamento de seu pipeline de dados.

### **Passos Rápidos para Configuração:**

* **Instale o Docker Desktop**: Disponível para [download](https://www.docker.com/products/docker-desktop/) no site oficial do Docker.
* **Organize as Pastas de Dados**: Estruture pastas locais para Bronze, Prata e Ouro para simular as camadas de armazenamento de dados.
* **Configure e Inicie o Airflow com Docker Compose**: Defina o serviço do Airflow no Docker Compose e inicie-o para acessar a interface web em http://localhost:8080.

<aside> 💡 **Dica**: Instale bibliotecas essenciais como **pandas** dentro do contêiner do Airflow para a manipulação de dados.

</aside>

## **Etapa 02) Criando o DAG no Airflow**

**Desenvolvimento do DAG:**

* Criar um DAG no Airflow que irá orquestrar todas as operações do pipeline de dados desde o carregamento até a transformação final.
* Definir tarefas sequenciais dentro do DAG para manipulação dos dados em cada camada.

<aside> 💡 **Dica:** Ao configurar o DAG, mantenha a legibilidade e a manutenibilidade em mente. Nomeie claramente cada tarefa e certifique-se de que as dependências entre as tarefas estão bem definidas para evitar ciclos e erros de execução.

</aside>

## **Etapa 03) Processamento e Limpeza de dados**

* **Carregar Dados Brutos na Camada Bronze:**
  + Implementar a função upload\_raw\_data\_to\_bronze para carregar dados brutos nos formatos CSV para a camada Bronze.
* **Limpeza de Dados para a Camada Prata:**
  + Utilizar a função process\_bronze\_to\_silver para ler e limpar os dados da camada Bronze:
    - Remover registros com campos nulos (nome, email, data de nascimento).
    - Corrigir formatos de email inválidos. (para ser um email valido é necessário ter o caracter “@”)
    - Calcular a idade dos usuários com base na data de nascimento.

<aside> 💡 **Dica**: Para as funções de limpeza de dados, considere usar expressões lambda e funções do pandas para eficiência. Por exemplo, ao verificar emails válidos, uma expressão lambda combinada com a função apply pode ser muito mais rápida do que um loop tradicional.

</aside>

## **Etapa 04) Transformação e Armazenamento de Dados**

1. **Transformações para a Camada Ouro:**
   * Com a função process\_silver\_to\_gold, ler os dados da camada Prata.
   * Executar transformações adicionais:
     + Agregar os dados por faixa etária e status (ativo ou inativo), facilitando análises demográficas e comportamentais.
     + Crie um dataset que mostre o número de usuários por faixa etária (0 a 10, 11 a 20, 21 a 30 anos…) e por status (“active” ou “inactive”)
   * Salvar os dados transformados na camada Ouro, prontos para análise e uso em decisões estratégicas.

<aside> 💡 **Dica:** Ao realizar operações de agregação como no processamento para a camada Ouro, use métodos otimizados da biblioteca **pandas** como **groupby** ou cut para classificar e agrupar dados. Essas funções são otimizadas internamente para lidar com grandes volumes de dados mais eficientemente do que implementações manuais.

</aside>

# **📝 Critérios de Avaliação**

Os critérios de avaliação mostram como você será avaliado em relação ao seu desafio.

| **Critérios** | **Atendeu às Especificações** | **Pontos** |
| --- | --- | --- |
| **Configuração e organização de pastas** | Verifica se as pastas de dados foram corretamente criadas e organizadas para simular as camadas de armazenamento (Bronze, Prata, Ouro). Avalia a clareza na estruturação e se as pastas estão configuradas para facilitar o acesso e a manipulação dos dados dentro do ambiente Docker. | **25** |
| **Camada Bronze** | Nessa etapa o aluno deverá implementar a função upload\_raw\_data\_to\_bronze para carregar os dados brutos. Verifica se os dados brutos são corretamente carregados na camada Bronze e se a integridade dos dados é mantida. | **25** |
| **Camada Prata** | Nessa etapa o aluno deverá realizar a eficácia das operações de limpeza de dados na camada Prata. Inclui a remoção de registros com campos nulos e a correção de e-mails inválidos. Verifica também a correta implementação das transformações como o cálculo da idade dos usuários. | **25** |
| **Camada Ouro** | Nessa etapa o aluno deverá realizar as transformações para a preparação dos dados para análises avançadas. Inclui a agregação dos dados por faixa etária e status e o correto armazenamento dos dados na camada Ouro, verificando se estão prontos para uso em decisões estratégicas. | **25** |

# **📆 Entrega**

<aside> ⚠️ **Atente-se a forma de nomear o REPOSITÓRIO: ele deve contar com o seu RID. Exemplo: (RID01234\_Desafio08). O RID pode ser encontrado dentro da sua plataforma em "meu perfil” e é composto por 5 números.**

</aside>

<aside> 📎 **Como entregar:** O aluno deve entregar o projeto através do link do repositório do Github ou um arquivo zipado.

</aside>

### **Atente-se ao formato de entrega deste desafio!**

1. Nomeie o seu colab com o RID e o número do desafio. Exemplo: RID1234\_Desafio01
2. Vá em Arquivo > Fazer download > Baixar o .ipynb
3. Faça o upload do arquivo no drive
4. Altere as configurações do arquivo para deixá-lo público.
5. Copie o link após alterar a permissão de acesso.
6. Submeta o link do arquivo (e não da pasta do drive!) na plataforma.